TF - LLM usage study with BigData, IoT, FOG

Objective:

- Evaluate LLM capability to assist Developer integrate his (her) application with BigData / IoT / FOG frameworks

Student: Henrique Krausburg Correa

Tutor: Prof. Dr.Cláudio Resin Geyer

Context

- Technology is growing faster and faster
  - From software perspective: new applications, frameworks, SDKs

- We seek to reuse solutions as much as possible, avoiding to “reinvent the wheel”
  - Thus, leading to more integration between systems

- However, read dozens of documentations implies spend more time that we are willing to
  - LLMs have been used exhaustively as assistants to resume information for us (among other tasks)

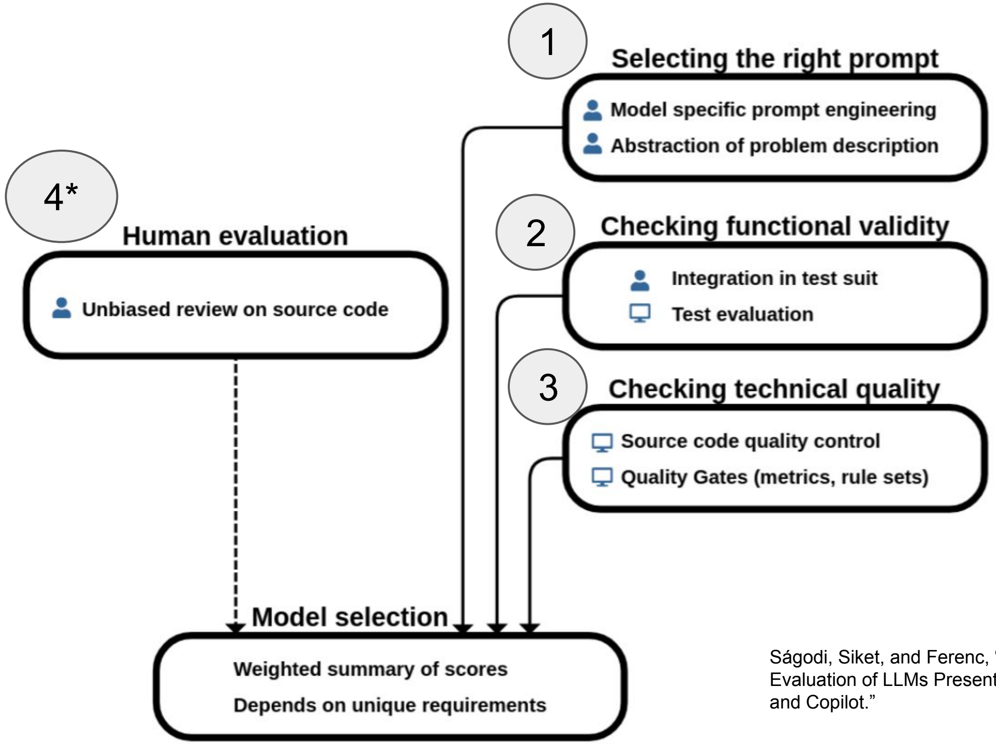
Context

- TL1 results
  - LLMs are excelling at specific tasks they were trained for
    - E.g.: Implement well known algorithms
  - Integrate different software domains imposes a challenge
    - E.g.: *“Implement my algorithm in a Hadoop cluster using Kubernets and Docker”*
    - Limited context for the LLM
      - Not know if trained with specific documentation
      - Too expensive to train models just to keep their knowledge base “up to date”
- Proposal
  - Explore methodology to choose the most suited LLM to assist the developer with software integration task

Methodology

- Apply paper methodology to choose LLM:
  - Ságodi, Zoltán, István Siket, and Rudolf Ferenc. "**Methodology for Code Synthesis Evaluation of LLMs Presented by a Case Study of ChatGPT and Copilot.**" *IEEE Access* (2024).
    - Authors propose methodology developers should use to choose the correct LLM model for code generation tasks.
    - Research questions:
      - **How does LLM-generated source code score in terms of source code quality?**
      - **Is the generated source code accepted by developers?**
      - **What aspects should be considered when choosing LLM-based generative tools?**

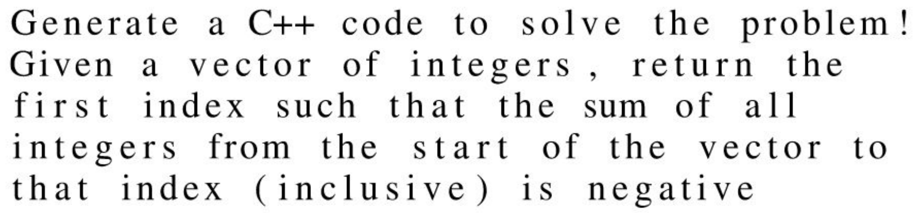
Methodology overview


1
Selecting the right prompt
Model specific prompt engineering
Abstraction of problem description
2
Checking functional validity
Integration in test suit
Test evaluation
3
Checking technical quality
Source code quality control
Quality Gates (metrics, rule sets)
4*
Human evaluation
Unbiased review on source code
Model selection
Weighted summary of scores
Depends on unique requirements

Ságodi, Siket, and Ferenc, “Methodology for Code Synthesis Evaluation of LLMs Presented by a Case Study of ChatGPT and Copilot.”

Paper - Selecting the right prompt

- Paper proposal: Build a collection of code challenge prompts
  - Solve challenges implementing C++ and Java applications
  - Zero shot prompts
- Takeaways
  - None of the models shall be favored by optimized prompting
  - Look for average difficult challenges
    - Easy assignments like “Hello world” provides no meaningful information
  - Do not give too much details about the problem
    - Won’t test model capability to identify connections between elements



```
Generate a C++ code to solve the problem!
Given a vector of integers, return the
first index such that the sum of all
integers from the start of the vector to
that index (inclusive) is negative
```

Ságodi, Siket, and Ferenc, “Methodology for Code Synthesis Evaluation of LLMs Presented by a Case Study of ChatGPT and Copilot.”

Paper - Checking functional validity

Achieved through functional testing (unit test)

- 25 tasks
  - Random cases: 10.000 test cases randomly selected from a pool of 1.000.000
  - Edge cases: from 5 to 40
- Authors state they did manual adaptations on generated code
  - Not always read input from correct interface (e.g.: console)

Paper - Technical quality

- Measure quality requirements through static analysis
- Authors used SonarQube and SonarScanner
- Metrics
  - Logical Lines of Code (LLOC)
  - Number of Statements (NOS)
  - McCabe Cyclomatic Complexity (McCC)
  - Nesting Level (NLE)

Paper - Human Evaluation

- Authors stated this phase as optional
    - Humans resource are expensive
- Reviewers must not be involved on previous phase
- Asked to evaluate source code in a even number scale
    - First Impression
    - Readability
    - Usability
    - Modifiability
- 5 Developers per language ranging 5 to 20 years industry experience

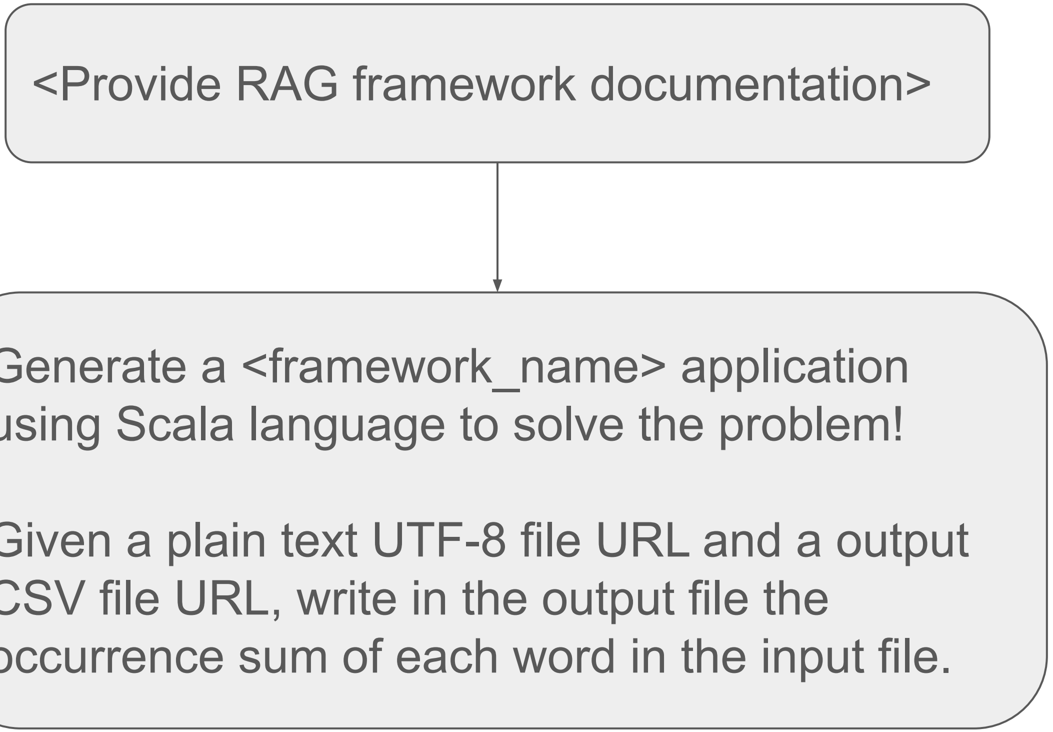
Solution architecture

- Use LLM to implement an application for a BigData Framework (TL1 approach)
- LLM candidates
  - Llama 3.1 70b instruct
  - Gemini 2.0 Flash Experimental
  - Phi 3 mini instruct
- Frameworks
  - Apache Spark
  - Apache Flink
- Language
  - Scala
- Application
  - word count
  - Input: Dracula by Bram Stoker

Source code: https://github.com/HenriKCorrea/llm-eval-for-hibench

Application - Create challenge prompt

- Prompt template



<Provide RAG framework documentation>

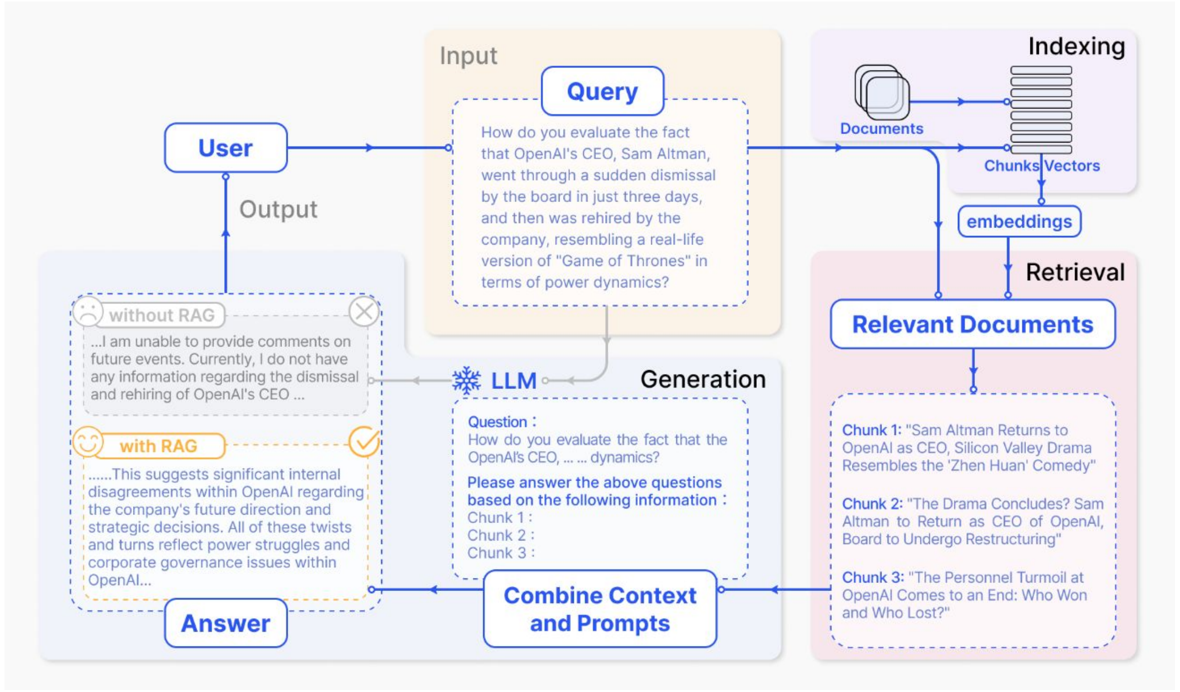
Generate a <framework_name> application using Scala language to solve the problem!

Given a plain text UTF-8 file URL and a output CSV file URL, write in the output file the occurrence sum of each word in the input file.

Application - Create challenge prompt

A representative instance of the RAG (Retrieval-Augmented generation) process applied to question answering. It mainly consists of 3 steps.

1. **Indexing**. Documents are split into chunks, encoded into vectors, and stored in a vector database.
2. **Retrieval**. Retrieve the Top k chunks most relevant to the question based on semantic similarity.
3. **Generation**. Input the original question and the retrieved chunks together into LLM to generate the final answer.



How do you evaluate the fact that OpenAI's CEO, Sam Altman, went through a sudden dismissal by the board in just three days, and then was rehired by the company, resembling a real-life version of "Game of Thrones" in terms of power dynamics?

...I am unable to provide comments on future events. Currently, I do not have any information regarding the dismissal and rehiring of OpenAI's CEO ...

......This suggests significant internal disagreements within OpenAI regarding the company's future direction and strategic decisions. All of these twists and turns reflect power struggles and corporate governance issues within OpenAI...

Question :
How do you evaluate the fact that the OpenAI's CEO, ... dynamics?
Please answer the above questions based on the following information :
Chunk 1 :
Chunk 2 :
Chunk 3 :

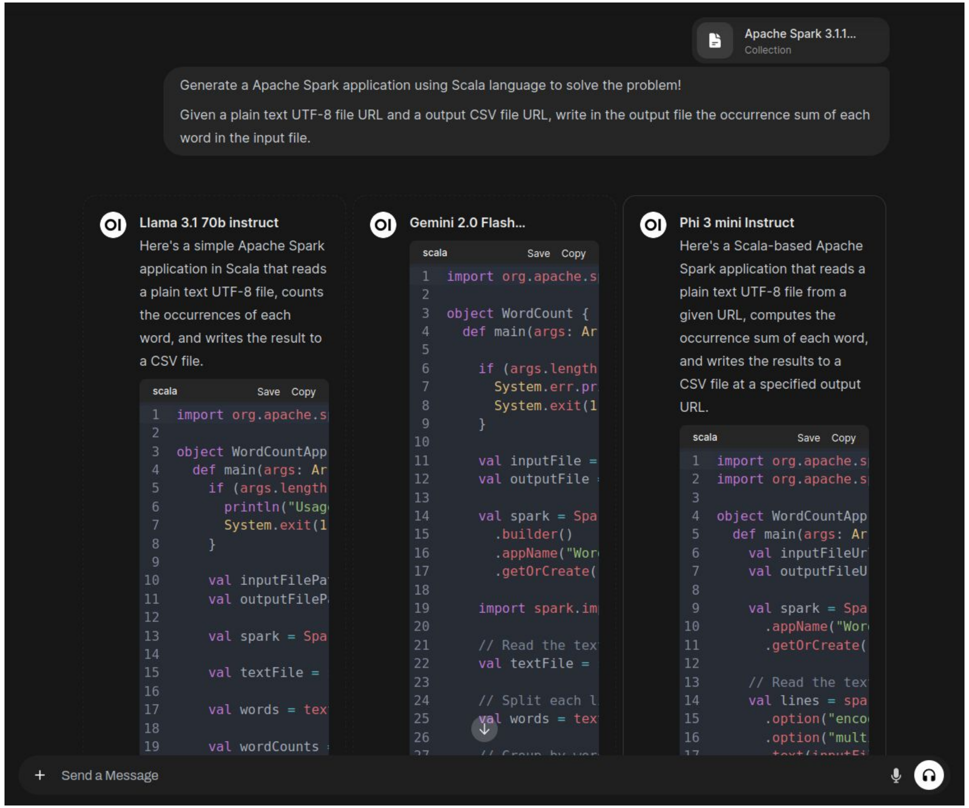
Chunk 1: "Sam Altman Returns to OpenAI as CEO, Silicon Valley Drama Resembles the 'Zhen Huan' Comedy"

Chunk 2: "The Drama Concludes? Sam Altman to Return as CEO of OpenAI, Board to Undergo Restructuring"

Chunk 3: "The Personnel Turmoil at OpenAI Comes to an End: Who Won and Who Lost?"

Gao et al., "Retrieval-Augmented Generation for Large Language Models."

Application - Create challenge prompt



Apache Spark 3.1.1...
Collection

Generate a Apache Spark application using Scala language to solve the problem!

Given a plain text UTF-8 file URL and a output CSV file URL, write in the output file the occurrence sum of each word in the input file.

Llama 3.1 70b instruct

Here's a simple Apache Spark application in Scala that reads a plain text UTF-8 file, counts the occurrences of each word, and writes the result to a CSV file.

```scala
import org.apache.s

object WordCountApp
  def main(args: Ar
    if (args.length
      println("Usag
      System.exit(1
    }

    val inputFilePa
    val outputFileP

    val spark = Spa

    val textFile =

    val words = tex

    val wordCounts
```

Gemini 2.0 Flash...

```scala
import org.apache.s

object WordCount {
  def main(args: Ar

    if (args.length
      System.err.pr
      System.exit(1
    }

    val inputFile =
    val outputFile

    val spark = Spa
      .builder()
      .appName("Wor
      .getOrCreate(

    import spark.im

    // Read the tex
    val textFile =

    // Split each l
    val words = tex
```

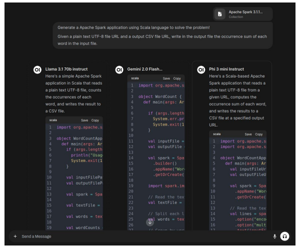
Phi 3 mini Instruct

Here's a Scala-based Apache Spark application that reads a plain text UTF-8 file from a given URL, computes the occurrence sum of each word, and writes the results to a CSV file at a specified output URL.

```scala
import org.apache.s
import org.apache.s

object WordCountApp
  def main(args: Ar
    val inputFileUr
    val outputFileU

    val spark = Spa
      .appName("Wor
      .getOrCreate(

    // Read the tex
    val lines = spa
      .option("enco
      .option("mult
```

Application - Functional validity

Prompt → Build → Run



Generate a Apache Spark application using Scala language to solve the problem!

Given a plain text UTF-8 file URL and a output CSV file URL, write in the output file the occurrence sum of each word in the input file.

Llama 3.1 70b instruct

Here's a simple Apache Spark application in Scala that reads a plain text UTF-8 file, counts the occurrences of each word, and writes the result to a CSV file.

```scala
import org.apache.s

object WordCountApp
  def main(args: Ar
    if (args.length
      println("Usag
      System.exit(1
    }

    val inputFilePa
    val outputFileP

    val spark = Spa

    val textFile =

    val words = tex

    val wordCounts
```

Gemini 2.0 Flash...

```scala
import org.apache.s

object WordCount {
  def main(args: Ar

    if (args.length
      System.err.pr
      System.exit(1
    }

    val inputFile =
    val outputFile

    val spark = Spa
      .builder()
      .appName("Wor
      .getOrCreate(

    import spark.im

    // Read the tex
    val textFile =

    // Split each l
    val words = tex
```

Phi 3 mini Instruct

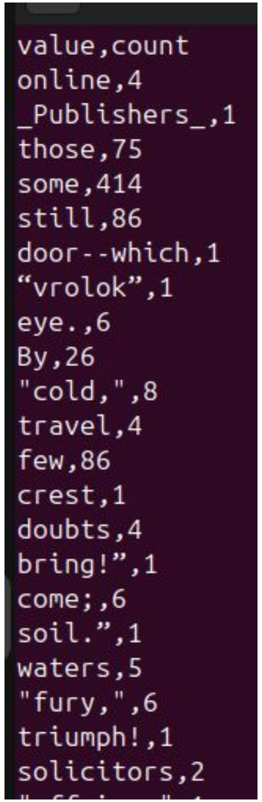
Here's a Scala-based Apache Spark application that reads a plain text UTF-8 file from a given URL, computes the occurrence sum of each word, and writes the results to a CSV file at a specified output URL.

```scala
import org.apache.s
import org.apache.s

object WordCountApp
  def main(args: Ar
    val inputFileUr
    val outputFileU

    val spark = Spa
      .appName("Wor
      .getOrCreate(

    // Read the tex
    val lines = spa
      .option("enco
      .option("mult
```

Application - Functional validity

WordCountGeminiApp.scala

- Compiled successfully.
  No intervention required.
- Code executed with success!

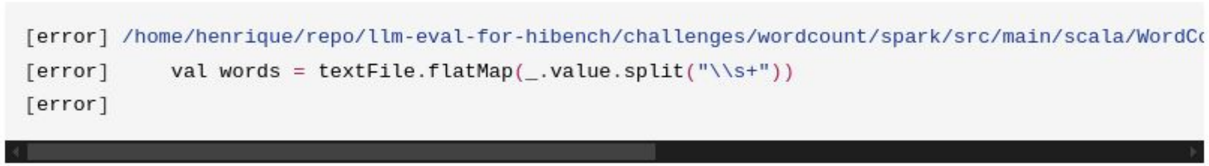


```
value,count
online,4
_Publishers_,1
those,75
some,414
still,86
door--which,1
“vrolok”,1
eye.,6
By,26
"cold,",8
travel,4
few,86
crest,1
doubts,4
bring!”,1
come;,6
soil.”,1
waters,5
"fury,",6
triumph!,1
solicitors,2
```

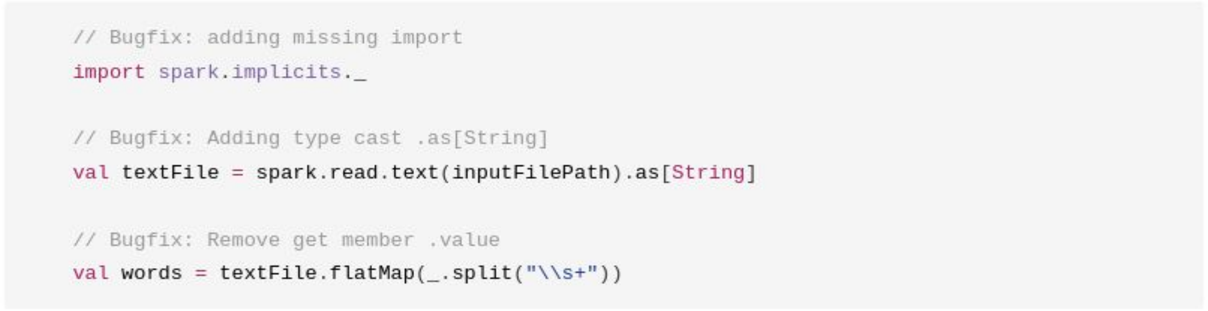
Application - Functional validity

WordCountLlamaApp.scala

- Build Issues:
  - Missing import
  - Missing type cast
  - Invalid member access



```
[error] /home/henrique/repo/llm-eval-for-hibench/challenges/wordcount/spark/src/main/scala/WordC
[error]     val words = textFile.flatMap(_.value.split("\\s+"))
[error]
```

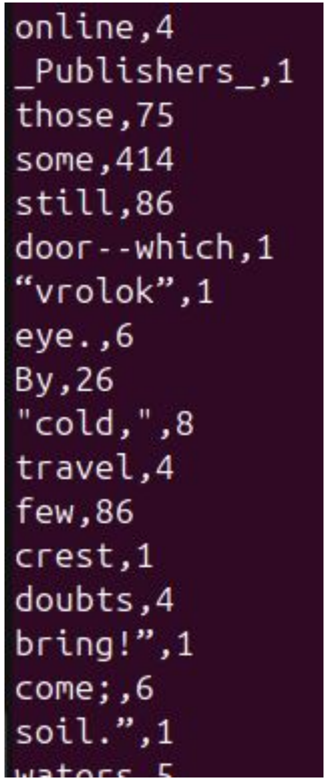


```
    // Bugfix: adding missing import
    import spark.implicits._

    // Bugfix: Adding type cast .as[String]
    val textFile = spark.read.text(inputFilePath).as[String]

    // Bugfix: Remove get member .value
    val words = textFile.flatMap(_.split("\\s+"))
```

Application - Functional validity

WordCountLlamaApp.scala

- Code executed successfully!

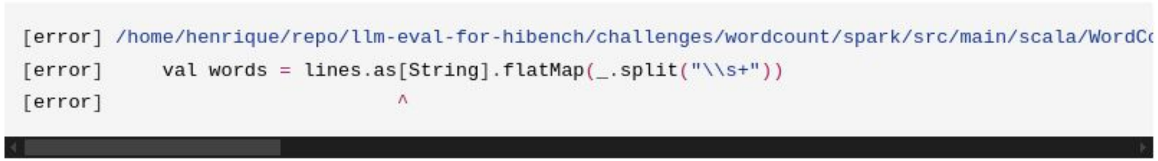


```
online,4
_Publishers_,1
those,75
some,414
still,86
door--which,1
“vrolok”,1
eye.,6
By,26
"cold,",8
travel,4
few,86
crest,1
doubts,4
bring!”,1
come;,6
soil.”,1
waters,5
```

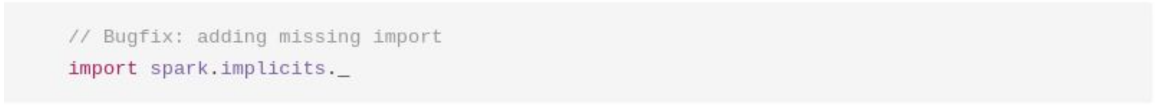
Application - Functional validity

WordCountPhiApp.scala

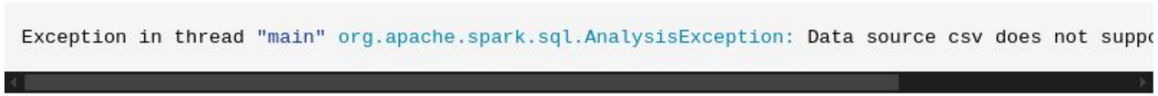
- Build Issues:
  - Missing import to support [String] encoder.
- Application failed



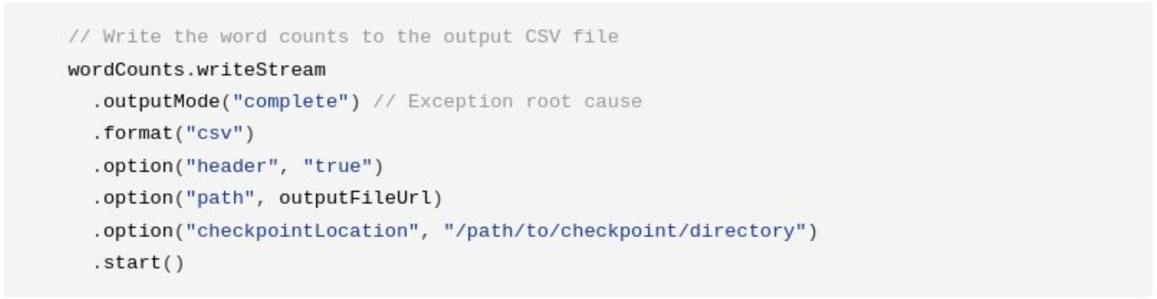
```
[error] /home/henrique/repo/llm-eval-for-hibench/challenges/wordcount/spark/src/main/scala/WordC
[error]     val words = lines.as[String].flatMap(_.split("\\s+"))
[error]                                  ^
```



```
    // Bugfix: adding missing import
    import spark.implicits._
```



```
Exception in thread "main" org.apache.spark.sql.AnalysisException: Data source csv does not supp
```

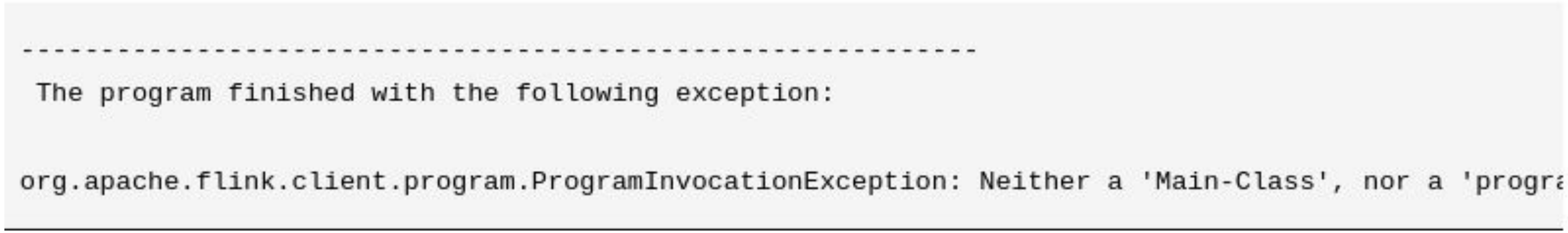


```
    // Write the word counts to the output CSV file
    wordCounts.writeStream
      .outputMode("complete") // Exception root cause
      .format("csv")
      .option("header", "true")
      .option("path", outputFileUrl)
      .option("checkpointLocation", "/path/to/checkpoint/directory")
      .start()
```

Application - Functional validity

WordCountGeminiFlinkApp.scala

- Compiled successfully. No intervention required.
- Application failed

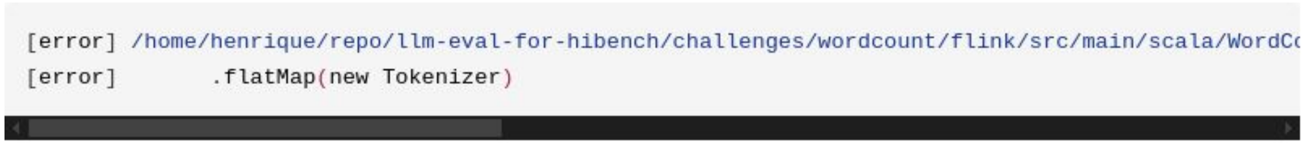


```
------------------------------------------------------------
 The program finished with the following exception:

org.apache.flink.client.program.ProgramInvocationException: Neither a 'Main-Class', nor a 'progra
```

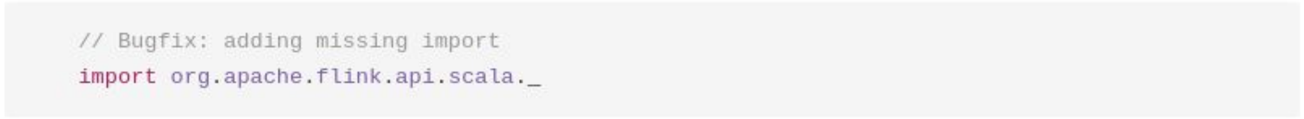
Application - Functional validity

WordCountLlamaFlinkApp.scala

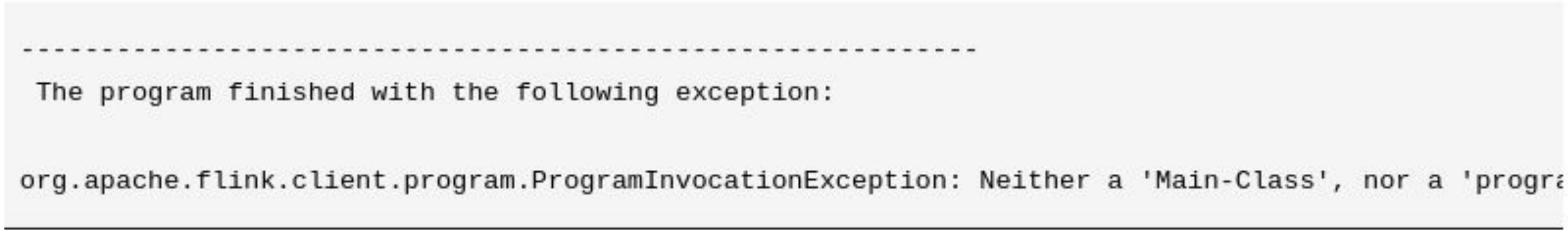
- Missing import to provide scala API return types.
- Application failed



```
[error] /home/henrique/repo/llm-eval-for-hibench/challenges/wordcount/flink/src/main/scala/WordC
[error]        .flatMap(new Tokenizer)
```



```
    // Bugfix: adding missing import
    import org.apache.flink.api.scala._
```

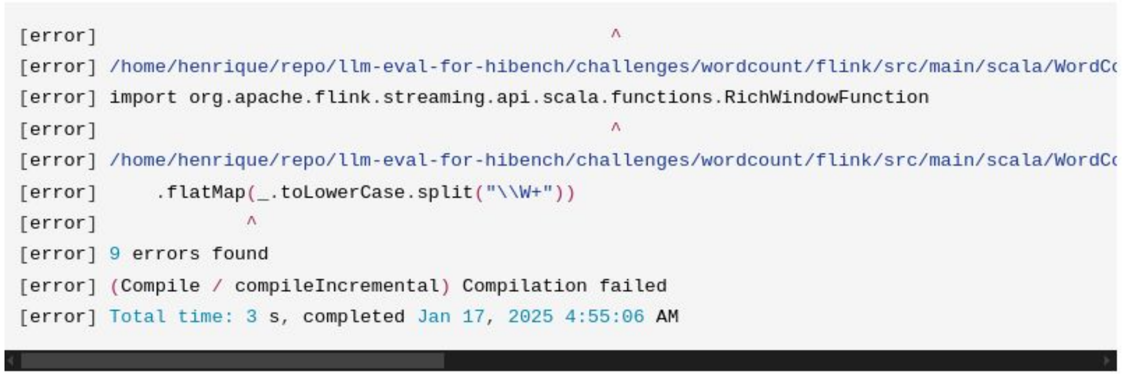


```
------------------------------------------------------------
 The program finished with the following exception:

org.apache.flink.client.program.ProgramInvocationException: Neither a 'Main-Class', nor a 'progra
```

Application - Functional validity

WordCountPhiFlinkApp.scala

- Build failed
  - Hallucination: generated many import and object members that does not exists

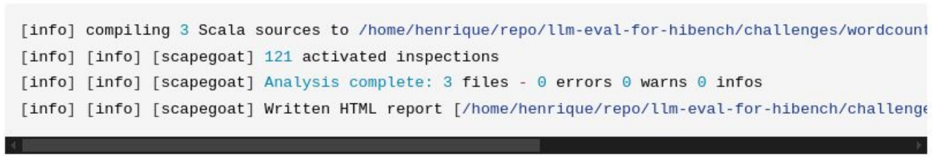


```
[error]                                                     ^
[error] /home/henrique/repo/llm-eval-for-hibench/challenges/wordcount/flink/src/main/scala/WordCo
[error] import org.apache.flink.streaming.api.scala.functions.RichWindowFunction
[error]                                                     ^
[error] /home/henrique/repo/llm-eval-for-hibench/challenges/wordcount/flink/src/main/scala/WordCo
[error]     .flatMap(_.toLowerCase.split("\\W+"))
[error]             ^
[error] 9 errors found
[error] (Compile / compileIncremental) Compilation failed
[error] Total time: 3 s, completed Jan 17, 2025 4:55:06 AM
```

Technical quality

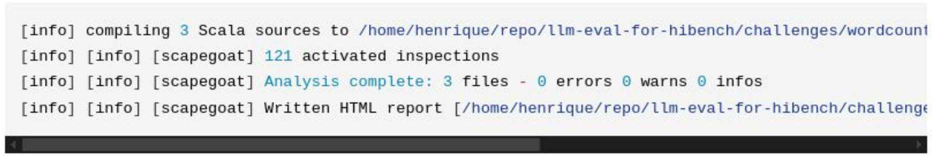
- Scapegoat has been used for Static analysis.
- No errors found on LLM generated source code.
- Over 123 inspections for Scala 2.
- See List of inspections for static analysis rules description.

Spark source code static analysis



```
[info] compiling 3 Scala sources to /home/henrique/repo/llm-eval-for-hibench/challenges/wordcount
[info] [info] [scapegoat] 121 activated inspections
[info] [info] [scapegoat] Analysis complete: 3 files - 0 errors 0 warns 0 infos
[info] [info] [scapegoat] Written HTML report [/home/henrique/repo/llm-eval-for-hibench/challenge
```

Flink source code static analysis



```
[info] compiling 3 Scala sources to /home/henrique/repo/llm-eval-for-hibench/challenges/wordcount
[info] [info] [scapegoat] 121 activated inspections
[info] [info] [scapegoat] Analysis complete: 3 files - 0 errors 0 warns 0 infos
[info] [info] [scapegoat] Written HTML report [/home/henrique/repo/llm-eval-for-hibench/challenge
```

Results

- Paper methodology applied successfully to Big Data / IoT context
  - Improvement: enhance LLM context according to real use case
  - Significant effort to build benchmark for specific use case
- Succeed run apache spark applications
  - Gemini 2.0 Flash Experimental showed best results
- Fail to apache flink applications
  - Documentation was not helpful?
  - Framework not mature enough?

Thanks!